**THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY**

**(Deemed to be University)**

**Patiala, Punjab**

****

**LAB ASSIGNMENT-1**

**Submitted By:**

**“Sajid Miya” 102367013**

**Subgroup: 2Q11**

**Submitted To:**

**THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY**

**Department of Computer Science and Engineering**

**Patiala, Punjab, India**

**“19th August 2024”**

UCS301 Data Structures Lab Assignment 1

(Week 1)

1. Develop a Menu driven program to demonstrate the following operations of Arrays

——MENU——- 1.CREATE

1. DISPLAY
2. INSERT
3. DELETE
4. LINEAR SEARCH
5. EXIT

Ans:

#include <iostream>

#include <cstdlib>

#include <typeinfo>

using namespace std;

void menu()

{

    cout << "1. Create array" << endl;

    cout << "2. Display array" << endl;

    cout << "3. Insert into array" << endl;

    cout << "4. Delete from array" << endl;

    cout << "5. Linear search in array" << endl;

    cout << "6. Exit the program" << endl;

}

void createArray(int \*&arr, int &size)

{

    cout << "Enter the size of the array: ";

    cin >> size;

    arr = new int[size];

    cout << "Enter " << size << " elements:" << endl;

    for (int i = 0; i < size; i++)

    {

        cin >> arr[i];

    }

}

void displayArray(int \*arr, int size)

{

    if (size == 0)

    {

        cout << "Array is empty!" << endl;

        return;

    }

    cout << "Array elements are: ";

    for (int i = 0; i < size; i++)

    {

        cout << arr[i] << " ";

    }

    cout << endl;

}

void insertIntoArray(int \*&arr, int &size)

{

    int pos, value;

    cout << "Enter position to insert (0-based index): ";

    cin >> pos;

    cout << "Enter value to insert: ";

    cin >> value;

    int newSize = max(size, pos + 1);

    int \*newArr = new int[newSize];

    // Copy elements up to the insertion position

    for (int i = 0; i < pos; i++)

    {

        newArr[i] = (i < size) ? arr[i] : 0; // Fill gaps with zeros

    }

    // Insert the new value

    newArr[pos] = value;

    // Copy remaining elements

    for (int i = pos + 1; i < newSize; i++)

    {

        newArr[i] = (i - 1 < size) ? arr[i - 1] : 0;

    }

    delete[] arr;

    arr = newArr;

    size = newSize;

}

void deleteFromArray(int \*&arr, int &size)

{

    int pos;

    cout << "Enter position to delete (0-based index): ";

    cin >> pos;

    if (pos < 0 || pos >= size)

    {

        cout << "Invalid position!" << endl;

        return;

    }

    int \*newArr = new int[size - 1];

    for (int i = 0; i < pos; i++)

    {

        newArr[i] = arr[i];

    }

    for (int i = pos + 1; i < size; i++)

    {

        newArr[i - 1] = arr[i];

    }

    delete[] arr;

    arr = newArr;

    size--;

}

void linearSearch(int \*arr, int size)

{

    int value;

    cout << "Enter value to search: ";

    cin >> value;

    for (int i = 0; i < size; i++)

    {

        if (arr[i] == value)

        {

            cout << "Value found at position " << i << endl;

            return;

        }

    }

    cout << "Value not found in the array" << endl;

}

int main()

{

    int \*arr = nullptr;

    int size = 0;

    int choice;

    while (true)

    {

        menu();

        cout << "Enter your choice: ";

        cin >> choice;

        switch (choice)

        {

        case 1:

            createArray(arr, size);

            cout << "Array created" << endl;

            break;

        case 2:

            displayArray(arr, size);

            break;

        case 3:

            insertIntoArray(arr, size);

            break;

        case 4:

            deleteFromArray(arr, size);

            break;

        case 5:

            linearSearch(arr, size);

            break;

        case 6:

            delete[] arr;

            cout << "Exiting program..." << endl;

            return EXIT\_SUCCESS;

        default:

            cout << "Invalid choice! Please try again." << endl;

        }

    }

            cin.ignore();

            cout << "Press Enter to continue...";

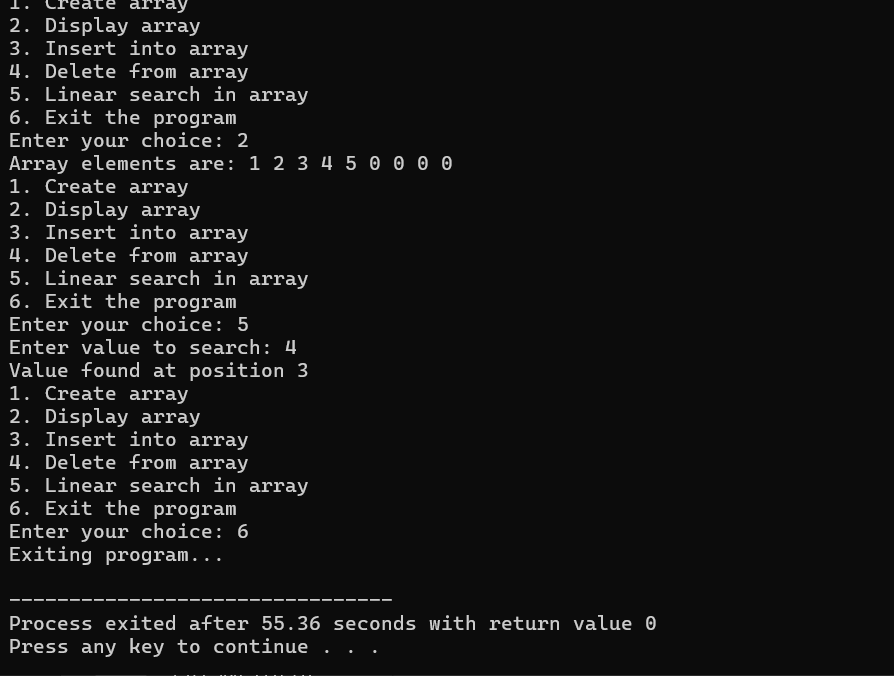
            cin.get();

    return EXIT\_SUCCESS;

}

Output:





1. Design the logic to remove the duplicate elements from an Array and after the deletion the array should contain the unique elements.

Ans:

#include<iostream>

using namespace std;

int main()

{

    //#ifndef ONLINE\_JUDGE

        //freopen("input.txt","r",stdin);

        //freopen("output.txt","w",stdout);

    //#endif

    int size;

    cout<<"enter the size of array : "<<endl;

    cin>>size;

    int arr[size];

    for (int i =0;i<size;i++)

    {

        cout<<"Element "<<i+1<<endl;

        cin>>arr[i];

    }

    int uniquearr[size];

    for (int i=0;i<size;i++)

    {

        for (int j=1;j<size-i-1;j++)

        {

            if (arr[j]>arr[j+1])

            {

                int temp= arr[j];

                arr[j]=arr[j+1];

                arr[j+1]=temp;

            }

        }

    }

    int k=0;

    int i;

    for ( i=0;i<size-1;i++)

    {

        if (arr[i]!=arr[i+1])

        {

            uniquearr[k++]=arr[i];

        }

    }

    cout<<"unique elements are : "<<endl;

    uniquearr[k++]=arr[i];

    for (i=0;i<k;i++)

    {

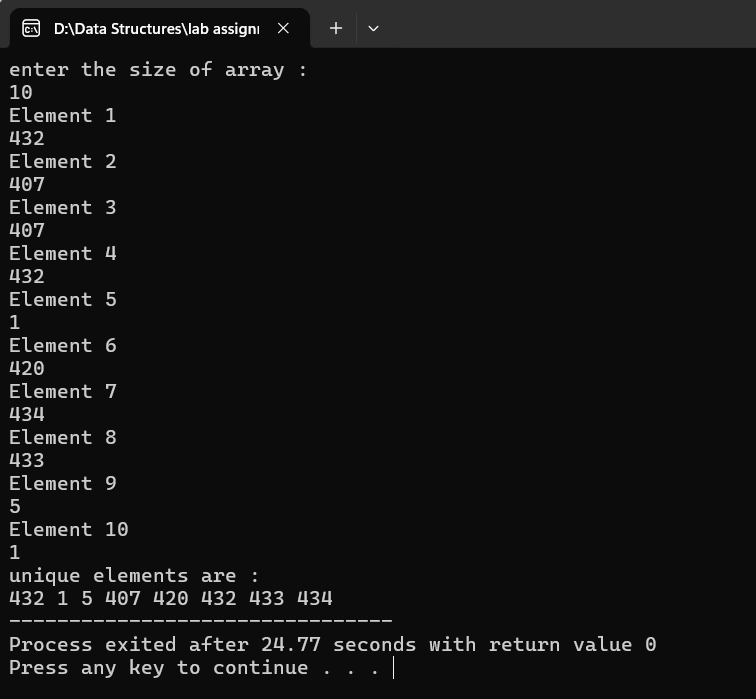
        cout<<uniquearr[i]<<" ";

    }

    return 0;

}

Output:



1. Predict the Output of the following program int main()

{

int i;

int arr[5] = {1};

for (i = 0; i < 5; i++) printf("%d",arr[i]);

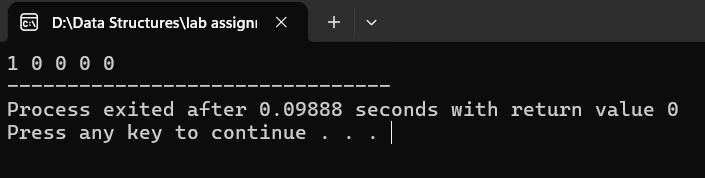
return 0;

}

Ans:

1 0 0 0 0

Output:



1. Implement the logic to
   1. Reverse the elements of an array

Ans:

#include<iostream>

using namespace std;

int main()

{

#ifndef ONLINE\_JUDGE

freopen("input.txt","r",stdin);

freopen("output.txt","w",stdout);

#endif

int size;

cout<<"enter the size of array ";

cin>>size;

int arr[size];

for (int i =0;i<size;i++)

{

cin>>arr[i];

}

cout<<"orignal array : "<<endl;

for (int i =0;i<size;i++)

{

cout<<arr[i]<<" ";

}

int range=size/2;

for (int i=0, j=size-1;i<range;i++,j--)

{

int temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

cout<<endl<<"reversed array : "<<endl;

for (int i =0;i<size;i++)

{

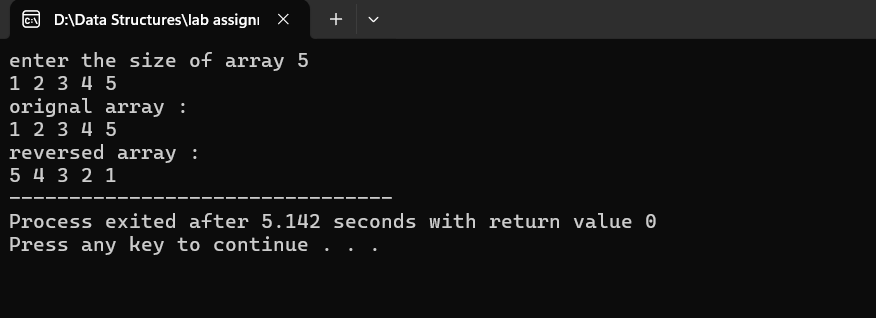
cout<<arr[i]<<" ";

}

return 0;

}

Output:



* 1. Find the matrix multiplication

Ans:

#include<iostream>

using namespace std;

int main(){

cout<<"enter the dimension of Matrix A : ";;

int row1,column1;

cin>>row1>>column1;

int A[row1][column1];

for (int i =0;i<row1;i++)

{

for(int j=0;j<column1;j++)

{

cout<<"A["<<i<<"]["<<j<<"]";

cin>>A[i][j];

}

}

cout<<"enter the dimension of Matrix B : ";;

int row2,column2;

cin>>row2>>column2;

int B[row2][column2];

for (int i =0;i<row2;i++)

{

for(int j=0;j<column2;j++)

{

cout<<"B["<<i<<"]["<<j<<"]";

cin>>B[i][j];

}

}

if (row2==column1)

{

int result[row1][column2];

for (int i=0;i<row1;i++)

{

for(int j=0;j<column2;j++)

{

result[i][j]=0;

for (int k=0;k<row2;k++)

{

result[i][j] += A[i][k] \* B[k][j];

}

}

}

for (int i=0;i<row1;i++)

{

for(int j=0;j<column2;j++)

{

cout<<result[i][j]<<" ";

}

cout<<endl;

}

}

else

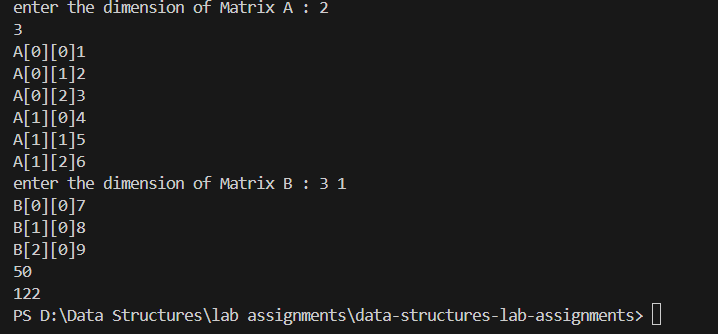
{

cout<<"matrix multiplication is not possible";

}

}

Output:



* 1. Find the Transpose of a Matrix

Ans:

#include<iostream>

using namespace std;

int main()

{

cout<<"enter the dimensions of the matrix ";

int row , column;

cin>>row>>column;

int arr[row][column];

for (int i =0;i<row;i++)

{

for(int j=0;j<column;j++)

{

cout<<"Arr["<<i<<"]["<<j<<"]";

cin>>arr[i][j];

}

}

cout<<"original matrix : "<<endl;

for (int i =0;i<row;i++)

{

for(int j=0;j<column;j++)

{

cout<<arr[i][j]<<" ";

}

cout<<endl;

}

int result[column][row];

for (int i =0;i<row;i++)

{

for(int j=0;j<column;j++)

{

result[j][i]=arr[i][j];

}

}

cout<<"Transposed matrix :"<<endl;

for (int i =0;i<column;i++)

{

for(int j=0;j<row;j++)

{

cout<<result[i][j]<<" ";

}

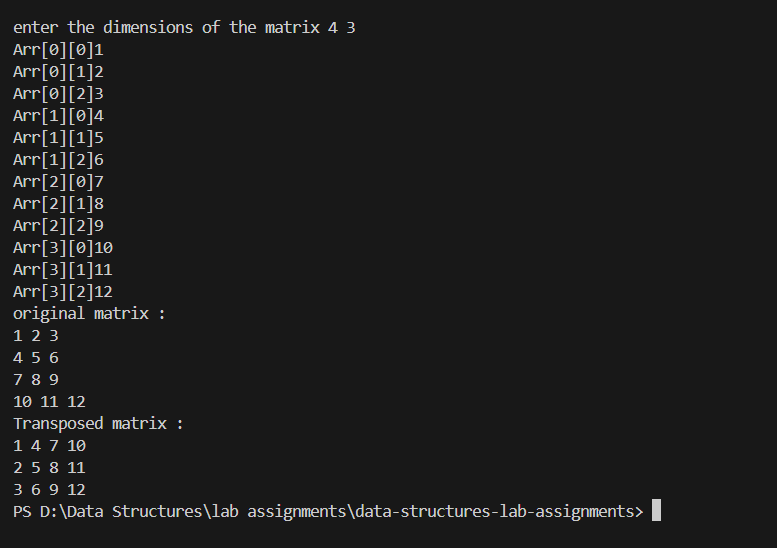
cout<<endl;

}

return 0;

}

Output:



1. Write a program to find sum of every row and every column in a two-dimensional array.

Ans:

#include<iostream>

using namespace std;

int main()

{

cout<<"enter the dimensions of the matrix ";

int row , column;

cin>>row>>column;

int arr[row][column];

for (int i =0;i<row;i++)

{

for(int j=0;j<column;j++)

{

cout<<"Arr["<<i<<"]["<<j<<"]";

cin>>arr[i][j];

}

}

for (int i =0;i<row;i++)

{

for(int j=0;j<column;j++)

{

cout<<arr[i][j]<<" ";

}

cout<<endl;

}

cout<<"sum of rows: ";

for (int i=0;i<row;i++)

{

int sum=0;

for (int j=0;j<column;j++)

{

sum+=arr[i][j];

cout <<arr[i][j]<<" ";

}

cout<<" sum of row "<<i+1<<" is "<<sum<<endl;

}

cout<<"sum of columns"<<endl;

for (int i=0;i<column;i++)

{

int sum=0;

for (int j=0;j<row;j++)

{

sum+=arr[j][i];

cout <<arr[j][i]<<" ";

}

cout<<" sum of column "<<i+1<<" is "<<sum<<endl;

}

}

Output:

